Simultaneous Identity and Expression Recognition using Face Geometry on Low Dimensional Manifolds

Pedro Martins and Jorge Batista *

Institute of Systems and Robotics
Dep. of Electrical Engineering and Computers
University of Coimbra - Portugal
{pedromartins,batista}@isr.uc.pt

Abstract. A solution for simultaneous identity and expression recognition is proposed. The proposed solution starts by extracting face geometry from input images using Active Appearance Models (AAM). Low dimensional manifolds were then derived using Laplacian EigenMaps resulting in two types of manifolds, one for model identity and the other for expression. Respective multiclass Support Vector Machines (SVM) were trained. The recognition is composed by a two step cascade, where first the identity is predicted and then its associated expression model is used to predict the facial expression. For evaluation proposes a database was build consisting on 6770 images captured from 4 people exhibiting 7 different emotions. The identity overall recognition rate was 96.8%. Facial expression results are identity dependent, and the most expressive individual achieves 76.8% of overall recognition rate.

Key words: Active Appearance Models, Laplacian EigenMaps, Support Vector Machines, Identity and Expression Manifolds.

1 Introduction

Facial expression is one of the most powerful, natural and immediate means for humans to share their emotions and intentions. Psychological studies focus on the interpretation on this mean to interact and describe that there are six basic emotions universally recognized [1], namely: joy, sadness, surprise, fear, anger and disgust. An automatic, efficient and accurate facial expression extraction system would thus be a powerfull tool assisting in these studies, allowing also other kinds of applications such as Human Computer Interface (HCI), smart interactive systems, video compression, etc. The proposed simultaneous identity and facial expression recognition it is based on the idea that it is straightforward for a human to capture the emotion and consequently the identity of a mimic actor our someone known using makeup. Humans can understand both the identity/expression based only in facial motion. This guidance idea lead to

* This work was funded by FCT grant SFRH/BD/45178/2008.

face geometry used to recognize the identity and facial expression (focusing on the six basic emotions plus the neutral one). Laplacian EigenMaps [2] are non-linear dimension reduction techniques that derive a low dimensional manifold lying in a higher dimensional more complex manifold. An identity/facial expression manifold is derived by embedding image data into a low dimensional space, where a image sequence is then represented as a trajectory in the parameter space. Learning a manifold of this nature require to derive a discriminative facial representation from raw images, in fact face images are represented by a set of sparse 2D feature point and the identity/expression manifolds were learned in a facial geometric feature space. The recognition has a feature extracting mechanism and a two stage cascade of multiclass Support Vector Machines (SVM) [3] classifiers trained with low dimensional manifold data of face geometry. Discriminative facial representation from raw images was achieved using Active Appearance Models (AAM) [4] that is an effective way to locate facial features, modeling both shape and texture from an observed training set, being able to extract relevant face information without background interference. For an input image, the AAM fitting framework extracts facial geometry related features, and the first SVM stage predict the identity, on the second SVM stage it is loaded the correspondent expression model for the predicted identity and the current expression is also predicted.

2 Active Appearance Models

Active Appearance Models (AAM) [4] are generative nonlinear parametric models of shape and texture, commonly used to model faces. These adaptive template matching methods, learn offline the variability of shape and texture, that is captured from a representative training set, being able to fully describe with photorealistic quality the trained faces as well as unseen.

2.1 Shape and Texture Models

The shape of an AAM is defined by the vertex locations of a 2D triangulated mesh. Mathematically, the representation used for a single v-point shape is a $2v$ vector given by $\mathbf{s} = (x_1, y_1, \ldots, x_v, y_v)^T$. The AAM training data consists of a set of annotated images with the shape mesh marked (usually by hand). The shapes are then aligned to a common mean shape using a Generalised Procrustes Analysis (GPA), removing location, scale and rotation effects. Principal Components Analysis (PCA) are then applied to the aligned shapes, resulting on the parametric model

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{n} p_i \mathbf{s}_i \tag{1}$$

where a new shapes, $\mathbf{s}$, are synthesised by deforming the mean shape, $\mathbf{s}_0$, using a weighted linear combination of eigenvectors, $\mathbf{s}_i$. n is the number of eigenvectors that holds a user defined variance, typically 95%. p_i is a vector of shape

parameters which represents the weights. Building a texture model, requires warping each training image so that the control points match those of the mean shape, $\mathbf{s}_0$. The texture mapping is performed, using a piece wise affine warp, i.e. partitioning the convex hull of the mean shape by a set of triangles using the Delaunay triangulation. Each pixel inside a triangle is mapped into the correspondent triangle in the mean shape using barycentric coordinates, see figure 1. This procedure removes differences in texture due shape changes, establishing a

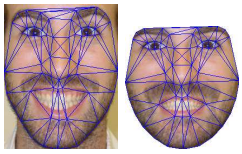


Fig. 1. On left the Original Image I, on right the Warped Image $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$

common texture reference frame. A texture model can be obtained by applying a low-memory PCA on the normalized textures. Defining pixel coordinates as $\mathbf{x} = (x, y)^T$, the appearance of the AAM is an image, $\mathrm{A}(\mathbf{x})$, defined over the pixels $\mathbf{x} \in \mathbf{s}_0$ such as $A(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \mathbf{A}_i(\mathbf{x}), \quad \mathbf{x} \in \mathbf{s}_0$. The appearance $\mathrm{A}(\mathbf{x})$ can be expressed as a base appearance $A_0(\mathbf{x})$ plus a linear combination of m appearance images $A_i(\mathbf{x})$ (EigenFaces). The coefficients λ_i are the appearance parameters.

2.2 Model Fitting

Fitting an AAM is usually formulated [5] as minimizing the texture error, in the least square sense, between the model instance $\mathrm{A}(\mathbf{x})$ and the input backwarped image onto the base mesh $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$,

$$\sum_{\mathbf{x}\in\mathbf{s}_0} \left[A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(\mathbf{W}(\mathbf{x},\mathbf{p})) \right]^2 . \tag{2}$$

In eq. 2 the warp $\mathbf{W}$ is the piecewise affine warp from the base mesh $\mathbf{s}_0$ to the current AAM shape $\mathbf{s}$, see figure 1. Hence, $\mathbf{W}$ is a function of the shape parameters $\mathbf{p}$. Notice that, the shape normalization on the model building process (Procrustes Analysis) the AAM do not model similarity transformations to the target image. Refer to [5] where is shown how to include it on the warp $\mathbf{W}(\mathbf{x};\mathbf{p})$.

The Simultaneous Inverse Compositional (SIC) [6] which minimize eq. 2 by performing a Gauss-Newtow gradient descent optimization simultaneously on the warp parameters $\mathbf{p}$ and the appearance parameters $\boldsymbol{\lambda}$ with respect to $\Delta\mathbf{p}$ and $\Delta\boldsymbol{\lambda}$, updating the warp by inverse composition: $\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$ and the appearance parameters additively: $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}$. Denoting,

$\mathbf{q} = \begin{pmatrix} \mathbf{p} \\ \boldsymbol{\lambda} \end{pmatrix}$, i.e. $\mathbf{q}$ is an $n+m$ dimensional vector containing the warp parameters $\mathbf{p}$ and the appearance $\boldsymbol{\lambda}$. The $m+n$ Steepest Descent images [6] are of the form

$$\mathbf{SD}_{SIC}(\mathbf{x}) = \left(\nabla A \frac{\partial \mathbf{W}}{\partial p_1}, \cdots, \nabla A \frac{\partial \mathbf{W}}{\partial p_n}, A_1(\mathbf{x}), \cdots, A_m(\mathbf{x}) \right) \tag{3}$$

where ∇A is defined as $\nabla A = \nabla A_0 + \sum_{i=1}^{m} \lambda_i \nabla A_i$. The parameters update is computed as

$$\Delta \mathbf{q} = -H_{SIC}^{-1} \sum_{\mathbf{x} \in \mathbf{s}_0} \mathbf{SD}_{SIC}^T(\mathbf{x}) E(\mathbf{x}), \quad H_{SIC} = \sum_{\mathbf{x} \in \mathbf{s}_0} \mathbf{SD}_{SIC}^T(\mathbf{x}) \mathbf{SD}_{SIC}(\mathbf{x}), \tag{4}$$

where H_{SIC} is the Gauss-Newtow approximation of the Hessian. The error image,$E(\mathbf{x})$, is defined as

$$E(\mathbf{x}) = I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - \left[A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) \right]. \tag{5}$$

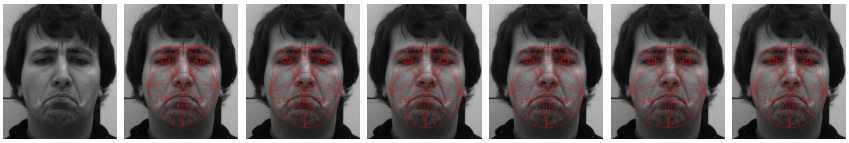
The Simultaneous Inverse Compositional when compared with other fitting approaches, such as the Project-Out [5] or the precomputed numerical estimate [4], work rather slow, since the Steepest Descent images depend on the appearance parameters and they have to re-computed in every iteration. By the other hand, SIC achieves the better fitting performance which is desirable for our proposes. Starting with a given estimate for the model, $\mathbf{q}_0$, and a rough estimate of the location of the face (provided by AdaBoost [7] method), an AAM model can be fitted with SIC following the algorithm 1. Figure 2 shows an example of AAM fitting into a target image.

Algorithm 1 Simultaneous Inverse Compositional Image Alignment

1: Evaluate the gradients ∇A_0 and ∇A_i for $i = 1, \cdots, m$
2: Evaluate the Jacobian of the warp $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x}; \mathbf{0})$
3: **while** MaxIterations reached or $|\Delta \mathbf{q}| < \varepsilon$ **do**
4: Warp I with $\mathbf{W}(\mathbf{x}; \mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$
5: Compute the error image, $E(\mathbf{x})$, using eq. 5
6: Compute the Steepest Descent images, $\mathbf{SD}(\mathbf{x})$, using eq. 3
7: Compute the Hessian matrix, H, eq. 4
8: Compute the parameters updates, $\Delta \mathbf{q}$, with eq. 4
9: Inverse Compose the Warp $\mathbf{W}(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; \Delta \mathbf{p})^{-1}$
10: Update the appearance parameters $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \Delta \boldsymbol{\lambda}$
11: **end while**

3 Laplacian EigenMaps

Laplacian EigenMaps [2] are nonlinear dimension reduction techniques that derive a low dimensional manifold lying in a higher dimensional more complex



(a) Input I (b) 1^{st} (c) 2^{nd} (d) 5^{th} (e) 8^{th} (f) 12^{th} (g) Final

Fig. 2. AAM fitting.

manifold. The Laplacian EigenMaps builds a graph that incorporates neighborhood information of the dataset and using the notion of the Laplacian of the graph, computes a low dimensional representation that optimally preserves local neighborhood information. Given k feature points $\mathbf{x}_1, \cdots, \mathbf{x}_k \in \Re^l$,a weighted graph with k nodes is build, one for each point, with a set of edges connecting neighboring points. The embedding map is found by computing the eigenvectors of the graph Laplacian [2]. See algorithm 2 where this method is described. Finding such embedding map, Φ, requires tuning n nearest neighbors for graph building and select the number of dimensions, m, where the input features were projected into.

Algorithm 2 Laplacian EigenMaps

- Build the Adjacency Graph:
Nodes i and j (or j and i) are connected by an edge to the n nearest neighbors.
- Choosing the weights W_{ij}: (if i and j are connected by an edge) then $W_{ij} = 1$
- Build EigenMaps:
Compute eigenvalues and eigenvectors for the generalized eigenvector problem

$$L\mathbf{f} = \lambda D\mathbf{f} \tag{6}$$

where $D_{ii} = \sum_j W_{ji}$ is a diagonal weight matrix and $L = D - W$ is the Laplacian matrix. Let $\mathbf{f}_0, \cdots, \mathbf{f}_{k-1}$ be the solutions of eq. 6 order by eigenvalues $\lambda_0 = 0 \leq \lambda_1 \leq \cdots \leq \lambda_{k-1}$). Leaving out the eigenvector $\mathbf{f}_0$ corresponding to eigenvalue 0, the embedding m-dimensional Euclidian space is given by $\Phi = [\mathbf{f}_1 | \mathbf{f}_2 | \cdots | \mathbf{f}_m]$.

4 Simultaneous Identity and Facial Expression Recognition

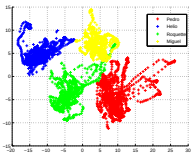
The proposed solution models both identity and facial expression in independent low dimensional manifolds. The system performs simultaneous identity and facial expression recognition by building different manifolds that were derived from embedding image data into a low dimensional subspace using Laplacian EigenMaps [2]. In order to learn these manifolds it is necessary to derive discriminative facial representation from raw images. This process it is done by the AAM fitting framework, see figure 2, where face images are represented by a set of sparse 2D feature point. As discriminatory features, insted of (x, y) feature points, were used AAM related geometric features, i.e. regarding eq. 1 the shape parameters, $\mathbf{p}$, provide the same geometric information but using lower dimensional features ($n << 2v$). All faces were normalized by selecting only shape

parameters that model only deformation (ignoring the 4 similarity parameters, refer to [5]). Both identity and expression manifolds were then learnt in a facial geometric feature space. One image sequence from a test subject describing a facial emotion is represented as a trajectory in the learnt manifold acquired from the parameter space of the AAM. See figure 4. These manifolds were build using Laplacian EigenMaps representations for the shape parameters (that are related to face geometry). This approach maps the dimensionality of $\mathbf{p}$ into a less dimensional space where the mapped features acquire a huge discrimination power. Two kinds of Laplacian EigenMaps were build. The first type of EigenMap (lets call it identity manifold) finds the lower dimensional manifold using data from all individuals, see figure 3. The second type, the expression manifold, uses data only from a single individual, that maps data emphasising the differences in individual facial motion of the different expressions, see figure 4. This system holds an indentity manifold and expression manifold for each of the individuals in the training set. For recognition proposes, several muticlass Support Vector Machines (SVM) [3] classifiers were build, where the low dimensional identity and expression manifolds, provide the training data in these models. Summarizing, the simultaneous identity/expression recognition has a feature extracting mechanism and a two stage cascade of SVM classifiers trained with embedded manifold data. For an input image, the AAM fitting framework extracts the normalized shape parameters, $\mathbf{p}$, and the first SVM stage predict the identity for these parameters. On the second SVM stage it is loaded the correspondent expression model for the predicted identity and the current expression is predicted also.

5 Experimental Results

For the purpose of this work, a Facial Dynamics Database was built. It consists of 4 individuals, in a frontal position, showing 7 different facial expressions, namely: neutral expression, happiness, sadness, surprise, anger, fear and disgust. All facial emotions were taken by starting and ending on the neutral expression. Each individual repeated all facial emotions four times. The dataset is formed by a total of 6770 images (640×480). The AAM model was build using a total of 28 images (7 images for each of the 4 person). Since the AAM will be used to fit every frame of the captured database, it should held as much shape variation possible. The training images were then composed by the most expressive images of the 7 emotions (from a random repetition sequence). These training images were hand annotated using 58 landmarks ($v = 58$). Training the model holding 95% of shape and appearance variance produces an AAM with 18 shape parameters, ($n = 18$), and 29 EigenFaces, ($m = 29$). All the 6770 frames of the Facial Dynamics Database were then fitted using the AAM model, retrieving the shape parameters, $\mathbf{p}$, for each frame. Two main schemes were used for the manifold building: setting data for identity and setting the data for the expressions of each individual. A total of 5 manifolds were constructed (one model for identity plus 4 for each person expressions). These Laplacian EigenMaps were build with

both the number of adjacency graph neighbours, and the number of dimensions where the input features were projected into, found by cross-validation. Figure 3 and 4 shows the manifolds produced for the identity and expressions respectively. Five multiclass SVM models were trained (again 1 for identity + 4 for expression). The multiclass SVM classication was achieved using one-against-all voting scheme with a Gaussian Radial Basis Function (RBF). The kernel parameters and the missclassification penalty, were found also by cross-validation. To evaluate the performance of the system the dataset was divided into 4 fold for cross validation F1, F2, F3 and F4, that matches to the 4 repetitions of all expressions that each subject was made. The results shown are confusion matrices that were obtained from the cross-validation of the 4 folds ([test F1, train F2,F3,F4]; [test F2, train F1,F3,F4;] ...). Identity and expression models were evaluated independently. Figure 3-left displays results for the identity recognition and table 1 shows results for the expression models for each person in the dataset. Regarding figure 4 it is noticed that person 1 (figure 4-most-left) is the most expressive. All facial emotions start and end from the neutral expression, which explains the high concentration of projected points over the neutral cluster. Experiments also shown that during the evolution of an emotion over time, due noise and the effect of confusion bettween expressions, the ground truth emotion is sometimes misclassified, i.e. the test point falls into other nearby cluster. This problem could be reduced by including facial dynamics constraints. Since our system only uses static based recognition, an improvement is expected by changing the way expressions are validated, that will be regarded in a near future work.



	Person1	**Person2**	**Person3**	**Person4**
Person1	**98.11**	0.09	1.79	0
Person2	1.32	**98.67**	0	0
Person3	2.93	0.29	**94.50**	2.27
Person4	1.29	0.13	2.32	**96.25**

Overall recognition rate = 96.88%

Fig. 3. Left - Identity manifold learnt with geometric AAM related features for 4 persons. Right - Confusion matrix for the identity manifold.

6 Conclusions

Simultaneous identity and expression recognition were achieved using a two stage classifier using high discriminative, low dimensional, geometric based features. Identity and expression of each individual were learn independently deriving a low dimensional manifold using Laplacian EigenMaps. Face geometric data was extracted using Active Appearance Models (AAM). For each image the AAM fitting framework provide normalized geometric related features and derived an identity manifold and expressions manifolds for each one of the individuals. Respective multiclass Support Vector Machines (SVM) were trained providing a two step classifier cascade where the first stage predicts identity. On second

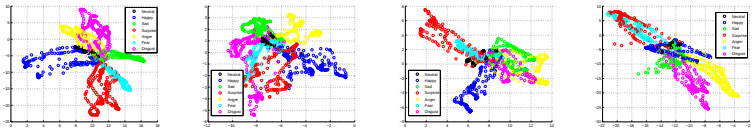


Fig. 4. Low dimensional manifolds learnt with geometric AAM related features for 4 persons exhibiting 7 expressions several turns each. Left-to-right figures represents the expression models for person 1, 2 ,3 and 4 respectively.

Table 1. Expression model confusion matrices for each one of the individuals.

Person 1

	Neut	Happ	Sad	Surp	Ang	Fear	Disg
Neut	**69.85**	9.16	2.29	0	0.76	1.14	16.79
Happ	0	**84.58**	3.33	10.41	1.66	0	0
Sad	0	0	**100**	0	0	0	0
Surp	0.66	0	0	**99.33**	0	0	0
Ang	2.3952	0	0.89	0.59	**84.43**	0.29	11.37
Fear	0	0.74	0	38.66	0	**60.59**	0
Disg	2.54	0	0	37.57	20.70	0	**39.17**

Overall recognition rate = 76.85%

Person 2

	Neut	Happ	Sad	Surp	Ang	Fear	Disg
Neut	**67.78**	0	6.37	0	0	25.83	0
Happ	1.14	**78.70**	0	17.11	0	3.04	0
Sad	1.73	0	**86.85**	5.53	0	0	5.88
Surp	0.76	25.95	0.76	**41.60**	0	26.71	4.19
Ang	1.38	0	1.84	0	**79.26**	0.46	17.05
Fear	2.86	0	2.04	57.37	0	**33.61**	4.09
Disg	1.62	17.26	3.58	22.80	1.62	2.93	**50.16**

Overall recognition rate = 62.56%

Person 3

	Neut	Happ	Sad	Surp	Ang	Fear	Disg
Neut	**43.71**	0	20.10	25.62	0	10.55	0
Happ	3.89	**80.52**	0.43	6.49	0	3.89	4.76
Sad	8.29	0	**72.48**	0	10.48	2.62	6.11
Surp	5.31	6.91	0	**65.95**	0	21.80	0
Ang	4.28	0.47	25.71	0	**61.90**	0.95	6.66
Fear	21.25	23.12	0	18.75	0	**23.13**	13.75
Disg	10.13	2.02	37.16	10.81	5.40	2.02	**32.43**

Overall recognition rate = 54.30%

Person 4

	Neut	Happ	Sad	Surp	Ang	Fear	Disg
Neut	**52.50**	17.50	0	18.00	0	0	12.00
Happ	4.67	**90.19**	0	3.73	0	1.14	0
Sad	2.01	12.56	**42.71**	0	0	0	42.71
Surp	1.86	2.80	0	**56.54**	0	32.71	6.07
Ang	2.19	0	0	0	**55.70**	0	42.10
Fear	1.43	3.34	0	16.26	0	**75.60**	3.34
Disg	0.49	6.46	0	0	0.99	0	**92.03**

Overall recognition rate = 66.47%

the expression model for the predicted identity is loaded and the expression is also predicted. For evaluation proposes an database was build having 6770 images captured from 4 people exhibiting 7 different emotions. Our 4 fold cross-validation results show that the system is able to recognize an overall 96.8% in the identity. The facial expression is very depend for each individual. In our dataset the most expressive individual achieves an overall recognition rate of 76.8% and the less expressive 54.3%.

References

1. *Handbook of Cognition and Emotion*, John Wiley & Sons Ltd, 1999.
2. Partha Niyogi Mikhail Belkin, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, 2003.
3. *The Nature of Statistical Learning Theory*, Springer-Verlag, N.Y., 1995.
4. G.J. Edwards T.F.Cootes and C.J.Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.
5. I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, 2004.
6. I.Matthews S. Baker, R.Gross, "Lucas kanade 20 years on: A unifying framework: Part 3," Tech. Rep., Carnegie Mellon University Robotics Institute, 2003.
7. Paul Viola and Michael Jones, "Rapid object detection using a boosted cascate of simple features," in *Computer Vision and Pattern Recognition*.